REMARKS

Reconsideration of the application is respectfully requested. Claims 1-7 are pending and remain in this application.

Drawings

The Office Action notes that the application was filed with informal drawings. Formal drawings will be filed upon receipt of a Notice of Allowance.

Rejections Under 35 U.S.C. § 102(e)

The Office Action has rejected claims 1-7 under 35 U.S.C. § 102(e) as being unpatentable over Wang. Applicant respectfully traverses this rejection. While Wang discloses a programmable logic device having programmable logic elements, it does not teach each and every claimed feature, which is required under Section 102(e). Firstly, it must be noted that Wang discloses a programmable logic device for use in an emulation system. Wang does *not* teach a method for compiling a user's design so that the design can be emulated in an emulator to eliminate timing problems.[1] Thus, the statements in the Office Action arguing that Wang teaches a method of compiling a netlist are incorrect.

In addition, contrary to the statements in the Office Action, Wang does not teach or suggest identifying every flip-flop in the emulation netlist, which is required by independent claims 1 and 4. All the Office Action says is that the logic element in Wang has a flip-flop. While this is a correct statement it misunderstands the difference between the flip-flop in Wang's logic element as opposed to the flip-flops of the "*emulation netlist*" of claims 1 and 4. Wang

[1] Wang only discusses the fact that rapid compilation of a user's design is desirable.

does not teach or suggest identifying every flip-flop in the "*emulation netlist*". As claim 1 makes clear, the "*emulation netlist*" is the netlist that is created after compiling the netlist that will be programmed into the emulator. In other words, the user of an emulation system creates a netlist description of an electronic design. Designers of such electronic systems need to functionally test these designs prior to manufacture and can use emulators for such tests. The methods of claims 1 and 4 compile the user's netlist to create the emulation netlist, which is the actual netlist that will be programmed into the emulation system.

In contrast, the emulator itself is comprised of integrated circuits like that disclosed in Wang that can have logic elements comprising flip-flops. While emulators using a programmable logic chip like that described in Wang have logic elements comprised of flip-flops, these flip-flops are not the flip-flops that form part of the emulation netlist, which represents the user's electronic circuit design.

Likewise, Wang does not teach or suggest changing the netlist of the circuit being tested to insert a variable delay element at the data inputs of the all of the flip-flops located in the emulation netlist. Just as the flip-flops in the emulation netlist are different than those that form a portion of the logic element of Wang, this variable delay element is not the same as that found in the logic element in Wang. Instead, the variable delay element of claims 1 and 4 are inserted into the emulation netlist, which as discussed is compiled from the netlist provided by the user. In contrast, the delay element in Wang is part of the logic element that is a building block for the integrated circuits used to build an emulator. [2] In sum, nothing in Wang teaches or suggests that the netlist of the design being tested should be compiled such that each flip-flop therein is

[2] Wang makes it clear that the logic element disclosed therein is a building block of a programmable logic chip that used to physically build the emulator. See e.g., Col. 1, lines 48-67 and Col. 2, line 66, through Col. 3, line 6.

identified and that a variable delay element be inserted at the data inputs to the identified flip-flops.

In order to more clearly show that the flip-flops and variable delay elements in claims 1 and 4 are different than those flip-flops and delay elements that might form part of the logic elements located on the emulation chips, claims 1 and 4 have been amended to make it more clear that the compiled emulation netlist is programmed into the emulator.

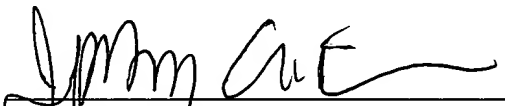
Claims 2-3 are allowable because they are dependent on claim 1, which is allowable. Likewise, claims 5-7 are allowable as well because they depend on claim 4. Applicant notes that the rejection of claim 7 actually demonstrates why all of the claims are allowable. Claim 7 calls for *removing* selected adjustable delay elements from the emulation netlist in certain specified conditions. As discussed above, the Office Action consistently argues that the delay element that form a portion of the logic elements disclosed in Wang are the same as the claimed adjustable delay elements. However, the delay element in Wang is *fabricated* into the programmable logic device, meaning that it *cannot* be removed, as is required by claim 7. That the delay element in Wang is hardwired into the chip is seen by reference to Fig. 9 of Wang, which is described as "a block diagram of the logical structure of the embodiment of *the emulation chip*". See Col. 4, lines 42-43. This block diagram shows "LEs", which are the logic elements illustrated in Fig. 11 and described at Columns 9-13. Given that the logic elements in Wang are fabricated into the emulation chip described in Wang, structures within that logic element such as the delay element cannot be removed from that logic element. Thus, Wang cannot possibly teach that the adjustable delay elements can be inserted and removed *since they are always present*! Applicant respectfully submits that Wang does not anticipate any of the claims in this application because

Wang does not teach anything about inserting or removing adjustable delay elements into or from an emulation netlist that will be programmed into an emulation system.[3]

Based upon the foregoing, Applicant respectfully submits that this application is in condition for allowance, which is respectfully requested. Should the Examiner have any questions or comments on the application, the Examiner should feel free to contact the undersigned via telephone.

Respectfully submitted,

ORRICK, HERRINGTON & SUTCLIFFE LLP

Dated: January 3, 2005 By: 

Jeffrey A. Miller
Reg. No. 35,287

Four Park Plaza, Suite 1600
Irvine, California 92614-2558
(650) 614-7660

[3] The Office Action notes that Figure 3 of the present application is the same as Figure 11 of Wang and then notes that the behavior of these circuits is inherently the same. This statement shows that the claimed methods are not understood. The claims describe a method of compiling a netlist so that the netlist can be programmed into an integrated circuit. The claims do not cover a circuit or even a circuit's behavior.